ArcMap .CAL Script: Spatial Join-Based Geometry Transfer

Contributed by Bert Granberg
04, Nov. 2009
Last Updated 05, Nov. 2009

Updated 11/5/09

You have a layer of polygons, say grazing permits, that are mostly based on another layer, say state and federal land ownership. Let's say the base data layer, land ownership, gets updated due to an improved survey. Is there an easy way to make the geometry adjustment to the grazing permits.

There are a couple of options to explore within the ArcMap UI, namely:
- Spatial Joins. Good approach but creates a new dataset as a result and sometimes the dataset + dataset = new dataset approach doesn't meet requirements and/or makes an unnecessary mess
- Spatial Adjustment toolbar --> Attribute Transfer tool, Geometry Transfer option. Also a good approach but involves mouse clicks, is not customizable, and doesn't resolve matching between single and multipart features

Here is another option, a .CAL script to run in the ArcMap field calculator on a selected set, within an edit session.

TO USE:
- open the attribute table for the target feature class (grazing in the example above), select one or more features (start with one at a time).
- Right click on the SHAPE field or hit CTRL + SHIFT + F to open the field calculator.
- In the field calculator, check the Advanced Option box and paste the script below in the Pre-Logic VBS Script Code text box.
- Make sure that the source layer number script parameter is set, This can be found by searching for SET THIS in the code. Set the layer index number for the polygon layer with the source geometry
- In the bottom box, under Shape =, type in: pOutPolygon

Notes:

If you're running this on a big selected set, the message box (shown below) will bug you. It can easily be disabled or deleted if desired. It's at the end of the script and can be disabled by sticking an apostrophe (') in front of it to comment that function out.

```
'
' .CAL Script Code Starts Here


' ** IMPORTANT SET THIS
' to the source polygon layer (i.e Cadastre.LandOwnership) position
' in the active ArcMap dataframes Table of Contents
Dim sourceGeometryTOCLayerIndex as Long
sourceGeometryTOCLayerIndex = 1    'note: layer 1 is the 2ND layer in the TOC

'** Get pointer variables to current arcmap project
Dim pMxDoc As IMxDocument
Dim pMap As IMap

Set pMxDoc = ThisDocument
Set pMap = pMxDoc.FocusMap

Dim pSourceGeometryLayer As IFeatureLayer
Set pSourceGeometryLayer = pMap.Layer(sourceGeometryTOCLayerIndex)

'** Set up a spatial filter and feature cursor to select
```

```
'** and iterate through contained polygon's that contain the
'** current grazing polygon's centroid
Dim pFCursor As IFeatureCursor
Dim pFeature As IFeature
Dim pGrazePolygon As IPolygon
Dim pGrazeArea As IArea
Dim currArea As Double
Dim currPartArea As Double
Dim pGrazeLabelPoint As IPoint
Dim pGrazeGC As IGeometryCollection
Dim pGrazePartGeometry As IGeometry
Dim pGrazePartPoly As IPolygon
Dim pGrazePartArea As IArea
Dim p As Long

Set pGrazePolygon = [Shape]
Set pGrazeGC = pGrazePolygon
Set pGrazeArea = pGrazePolygon 'QI
currArea = pGrazeArea.Area

Dim pOutPolygon As IGeometryCollection
Dim pOutArea As IArea
Dim pOwnGC As IGeometryCollection
Dim ownPart As Long
Dim pownArea As IArea

Set pOutPolygon = New Polygon
Set pOutArea = pOutPolygon

'Loop through each part of existing shape
For p = 0 To pGrazeGC.GeometryCount - 1

   Set pGrazePartGeometry = pGrazeGC.Geometry(p)
   If TypeOf pGrazePartGeometry Is IArea Then
      Set pGrazePartArea = pGrazePartGeometry
      currPartArea = pGrazePartArea.Area

      Set pGrazeLabelPoint = pGrazePartArea.LabelPoint

 'Spatial Query at Centroid
      Dim pSpatialFilter As ISpatialFilter
      Set pSpatialFilter = New SpatialFilter
      Set pSpatialFilter.Geometry = pGrazeLabelPoint
      pSpatialFilter.SpatialRel = esriSpatialRelWithin

      Set pFCursor = pSourceGeometryLayer.Search(pSpatialFilter, True)
      Set pFeature = pFCursor.NextFeature


      Do Until pFeature Is Nothing
         Set pOwnGC = pFeature.Shape
         Set pownArea = pOwnGC

         'Loop through each part in source polygon Geometry
         For ownPart = 0 To pOwnGC.GeometryCount - 1
            pOutPolygon.AddGeometry pOwnGC.Geometry(ownPart)
         Next ownPart
         Set pFeature = pFCursor.NextFeature

      Loop
   Else
      MsgBox "Wrong geometry type, check layer numbers under SET THIS"
   End If
Next p
```

```
Dim pTopOp As ITopologicalOperator
Set pTopOp = pOutPolygon
pTopOp.Simplify

msgbox "Object ID = " & [OBJECTID] & vbnewline & _
      "Area Before: " & currArea & vbnewline & "Area After:  " & pOutArea.Area _
      & vbnewline & "New Area as % of Old Area = " &  cstr(pOutArea.Area / currArea * 100)
```